ROBOFUN.RO

LECȚIA VII

CURS GRATUIT

ARDUINO ȘI ROBOTICĂ

Senzori distanţă

Proiect :

Harpă cu laser

Senzori de Distanta

Senzorii de distanta sunt capabili sa spuna cate de departe este obiectul din fata lor. In functie de principiul constructiv, exista mai multe categorii de senzori.

Avem astfel senzorii care se bazeaza pe emiterea de ultrasunete si masurarea timpului necesar ca sa se intoarca ecoul (principiul pe care functioneaza si navigatia liliacului). Acestia sunt senzori destul de precisi, foarte usor de folosit si a caror iesire variaza direct proportional cu distanta masura (un obiect situat la 2 metri va da un semnal de iesire de doua ori mai mare decat un obiect situat la 1 metru). Din cauza faptului ca sunetul se deplaseaza cu o viteza fixa, aceasta categorie de senzori este relativ lenta (in sensul ca daca vrem sa facem 100 de determinari intr-o secunda, acesti senzori nu vor fi capabili sa faca asta). In aceasta categorie se incadreaza sonarele MaxBotics si senzorul tip PING)).

A doua categorie de senzori sunt cei bazati pe reflexia unei raze de lumina infrarosie. Acesti senzori au doua zone active, o zona care emite lumina si o zona care receptioneaza raza reflectata de obiectul pana la care dorim sa masuram distanta. In functie de unghiul sub care se reflecta raza de lumina se poate determina distanta pana la obiect.

Acesti senzori sunt mult mai rapizi decat cei ultrasonici, insa functioneaza corect doar intr-o gama mai stricta de distante. Astfel avem un tip de senzor infrarosu in gama 3 – 40 cm, un alt tip in gama 10 – 80 cm si un alt tip in gama 15 – 150 cm. Mai exista si doua tipuri de senzori digitali, unul de 5 cm si unul de 10 cm. Senzorii digitali determina daca exista un obiect la o distanta mai mica de 5, respectiv 10 cm in fata senzorului. Pentru senzorul de tip 10-80 cm, valoarea citita de Arduino pe portul analogic la care este conectat senzorul va fi aproape de zero atunci cand nu exista nici un obiect in fata senzorului si aproximativ 630 atunci cand obiectul este la 10 cm in fata senzorului. Daca senzorul se apropie si mai mult de obiect (astfel ca distanta devine mai mica de 10 cm), valoarea citita scade iarasi, ajungand sa fie in jur de 430 cand senzorul este la cativa milimetri de obiect. Din acest motiv, daca avem nevoie de o determinare exact intr-o gama mai larga de distante, o solutie buna este sa utilizezi o combinatie de doi sau mai multi senzori, in functie de ce ai nevoie. Spre exemplu, ca sa faci un robot care ocoleste obstacole, un singur senzor 10-80 cm este suficient (cand obiectul ajunge la mai putin de 15 cm, faci robotul sa-si schimbe directia si sa-l ocoleasca. Pentru un robot de sumo insa, unde este important sa stii cand adversarul a ajuns la 2 cm de tine, vei vrea sa combini un senzor de 10-80 cm cu un senzor digital de 10 cm. Astfel vei folosi senzorul digital de 10 cm ca sa iti spuna daca adversarul este la mai putin de 10 cm de tine, si senzorul de 10-80 ca sa determini exact distanta. Senzorii Sharp sunt unii dintre cei mai des folositi senzori, intrucat sunt o combinatie echilibrata intre pret si performanta.

In sfarsit, a treia categorie de senzori sunt senzorii de tip laser. Acestia sunt cei mai precisi si cei mai rapizi, dar pot costa cu un ordin sau doua de marime fata de cele doua categorii anterioare (in gama sutelor sau miilor de euro pe bucata).

Sharp 3-40 cm, Sharp 10-80 cm si Sharp 15-150 cm

Acesti trei senzori sunt extrem de similari, difera doar gama distantelor in care sunt utili. Conectarea la Arduino este identica, iar codul folosit este aproape identic.

Pentru ca mufa de conectare este mai complicat de folosit altfel, recomand achizitionarea unui cablu special JST, care intra direct in mufa si ofera ca iesire 3 fire (ROSU – alimentare – VCC Arduino, NEGRU – masa – GND Arduino, si ALB – semnal – un pin analogic Arduino).

Imediat dupa ce ai conectat senzorul, urmatorul pas ar fi sa vezi valorile date de senzor, fara a mai aplica nici o procesare suplimentara. Pentru aceasta, incarca pe placa Arduino un program care afiseaza valorile senzorului, ca mai jos.

```
void setup() {
 Serial.begin(9600);
}
void loop() {
  int valoareSenzor = analogRead(0);
  Serial.print("Valoare Senzor: ");
  Serial.println(valoareSenzor,DEC);
}
```

Deschide Serial Monitor (Tools -> Serial Monitor) si apropiind si departand mana de senzor urmareste ce se intampla cu valorile citite. Pentru toti senzorii vei observa ca atunci cand nu ai nici un obiect in fata senzorului valoarea citita este mica, in zona 10-20-30. Pe masura ce mana ta se apropie de senzor, valoarea incepe sa creasca, pana la aproximativ 630 cand mana ta ajunge la 10 cm de senzor (valoare pentru senzorul 10 – 80 cm). Daca apropii mana si mai mult, valoarea incepe iarasi sa scada.

Pentru marea majoritate a proiectelor, acest cod este suficient. Sigur, te intrebi probabil "si unde e distanta, ca deocamdata sunt doar niste numere ???" . Asa este, ai dreptate. Cea mai simpla varianta ar fi sa-ti etalonezi singur cateva valori care te intereseaza (spre exemplu, masori experimental ca la distanta de 25 de cm ai o valoare citita de 432). O varianta ceva mai complicata este sa analizezi graficul valorilor din datasheet-urile senzorilor si sa faci o determinare matematica prin calcule, alegand puncte de referinta din grafic (folosind de exemplu metoda celor mai mici patrate). De remarcat ca pentru fiecare dintre senzori caracteristica (modul cum variaza valoarea citita de Arduino cu distanta pana la obiect) este diferita.

Pentru senzorul 10-80 cm spre exemplu, functia matematica de mai jos (determinata matematic plecand de la caracteristica din datasheet) aproximeaza distanta in centimetri in functie de valoarea citita de Arduino (este valabila doar pentru senzorul 10-80 cm).

```
int readDistance() {
   float volts = analogRead(0)* ((float) 5 / 1024);
   float distance = 65*pow(volts, -1.10);
   return distance;
}
```

Datorita faptului ca emit lumina infrarosie foarte des, in mod automat senzorii Sharp sunt

destul de mari consumatori de curent si sunt afectati de perturbatii in sursa de alimentare (cum ar fi, de exemplu, situatia unui robot alimentat cu o baterie mai putin capabila – sa spunem o baterie dreptunghiulara de 9 V – si la care atat motoarele cat si senzorii merg pe aceeasi baterie). Intr-o situatia de acest gen, valorile indicate de senzori pot fi mai mult sau mai putin eronate. Pentru a evita acest lucru, este bine ca intotdeauna senzorii sa fie alimentati separat (dintr-o alta baterie sau sursa de energie) decat motoarele sau alti mari consumatori de energie. Alta abordare este sa faci medierea valorilor citite de la senzor, ca mai jos.

```
int readDistanceMediata() {
 int sum = 0;
 for (int i=0; i<10;i++){
   float volts = analogRead(0)* ((float) 5 / 1024);
   float distance = 65*pow(volts, -1.10);
   sum = sum + distance;
   delay(5);
 }
 return (int)(sum / 10);
}
```

In acest exemplu am ales sa mediez 10 valori ale senzorului, lucru care ajuta foarte mult pentru situatiile cand valorile citite de la senzor nu sunt stabile. Evident, poti alege sa mediezi un alt numar de valori. Cu cat mediezi mai multe valori, cu atat senzorul este mai stabil, dar mai lent. Cu cat mediezi mai putine, cu atat senzorul este mai rapid.

Alte abordari pentru a face ca citirile senzorilor sa fie cat mai exact (abordari la nivel electric, de data aceasta) sunt sa adaugi in circuit unul sau doi condensatori, cu rolul de a netezi fluctuatiile tensiunii de alimentare.

De ce nu merge ?

–Asa cum spuneam si mai sus, senzorii Sharp sunt mari consumatori de curent. Daca sursa ta de alimentare nu este suficient de capabila, atunci valorile citite de senzori nu vor fi corecte. Spre exemplu, daca ai Arduino alimentat doar prin USB, atunci vei obtine valori mai mici decat cele reale. De asemenea, daca ai alimentat din aceeasi sursa de tensiune si motoarele robotului tau, si Arduino, iar sursa ta de alimentare nu este foarte capabila, atunci vei observa ca valorile citite de senzori sunt afectate de activitatea motoarelor (spre exemplu, cand motoarele pleaca de pe loc, acestea consuma mult curent, lasand senzorii fara curent). Exista mai multe solutii pentru a corecta aceasta problema. Cea mai simpla este ca intotdeauna sa alimentezi Arduino separat si motoarele separat, folosind o sursa de alimentare capabila (in mod normal, 6 baterii R6 de 1.5V fiecare, sau o baterie LIPO reprezinta surse de alimentare capabile; o baterie patrata de 9V insa, NU este niciodata o sursa capabila). Alta varianta este ca pur si simplu sa mediezi valorile intoarse de senzori (vei avea o latenta in citiri, dar erorile vor fi diminuate). In sfarsit, cea de-a treia solutie este din zona hardware si presupune conectarea in paralel cu alimentarea senzorului a unui condensator electrolitic de cativa zeci de microfarazi. Daca vrei sa aprofundezi aceasta abordare, iti recomand o cautare folosind Google dupa "sharp sensor capacitor filtering".

Sharp digital 10 cm, Sharp digital 5 cm

Senzorii digitali scot pe iesire valoarea 0 daca nu exista nici un obiect in raza lor si 1 daca exista un obiect. Conectarea la Arduino este la fel de simpla ca si in cazul senzorilor de mai sus. Pinul VCC se conecteaza la pinul 5V al Arduino, pinul GND se conecteaza la pinul GND al Arduino, iar pinul de semnal se conecteaza la un pin digital al Arduino. Codul sursa este foarte simplu, ca mai jos (unde am considerat ca senzorul este conectat pe pinul digital 7).

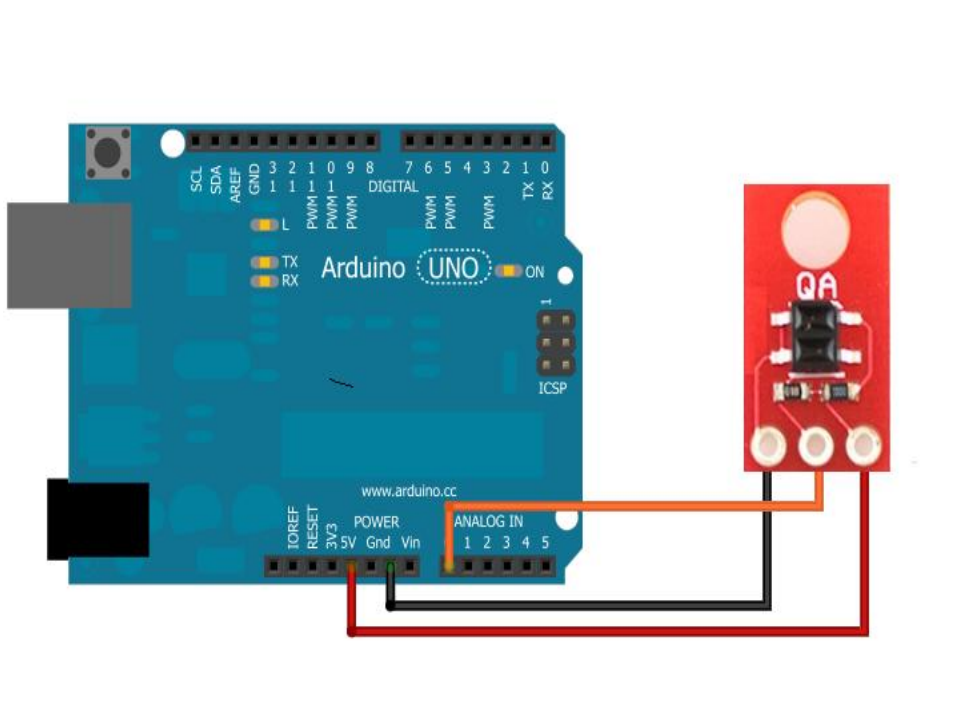
```
void setup() {
 Serial.begin(9600);
  pinMode(7, INPUT);
}

void loop() {
  int valoareSenzor = digitalRead(7);
  Serial.print("Valoare Senzor: ");
  Serial.println(valoareSenzor,DEC);
}
```

Senzor de Linie Analogic

Senzorul de linie este capabil sa detecteze gradul de reflectivitate pentru suprafata din fata senzorului. Ca aplicatie imediata, tinand cont de faptul ca o zona de culoare neagra reflecta foarte putin, iar o zona de culoare alba reflecta foarte puternic, acest senzor este utilizat in mod deosebit in robotica, pentru a permite unui robot sa faca distinctia intre suprafete de culoare neagra si suprafete de culoare alba. Acest lucru este util, spre exemplu, in cadrul concursurilor de sumo robotic, unde ringul de culoare neagra este marginit de o banda de culoare alba, sau pentru a face ca un robot sa urmareasca o linie de culoare neagra, pe fundal alb.

Ca principiu constructiv, senzorul consta intr-un led infrarosu si un receptor infrarosu, montati unul langa celalalt. Periodic (in mod automat, fara a fi controlat de tine in vreun fel) led-ul infrarosu emite lumina, lumina care este reflectata de suprafata din fata senzorului. Receptorul infrarosu culege lumina infrarosie reflectata de suprafata, si ofera pe pinul OUT o tensiune proportionala cu nivelul de lumina reflectata.

Astfel, conectarea la Arduino este foarte simpla. Senzorul trebuie alimentat (pinul VCC la pinul 5V al Arduino, pinul GND la pinul GND al Arduino, si pinul OUT la unul dintre pinii analogici Arduino – sa alegem A0 pentru programul de mai jos).



Arduino 5V	PIN1 senzor
Arduino GND	PIN1 Rezistor 10K
PIN2 Rezistor 10K	Pin2 Senzor
Arduino Analog0	PIN2 Rezistor 10K

```
void setup() {
 Serial.begin(9600);
}
void loop() {
  int nivelReflectanta = analogRead(0);
  Serial.print("Nivel Reflectanta: ");
  Serial.println(nivelReflectanta,DEC);
}
```

Senzori Prezenta Umana

Senzorii de prezenta umana ("PIR", de la :"Passive Infrared") detecteaza miscarea corpurilor vii in zona senzorului. Probabil ca stii deja acesti senzori (spre exemplu, cam toate toaletele benzinariilor

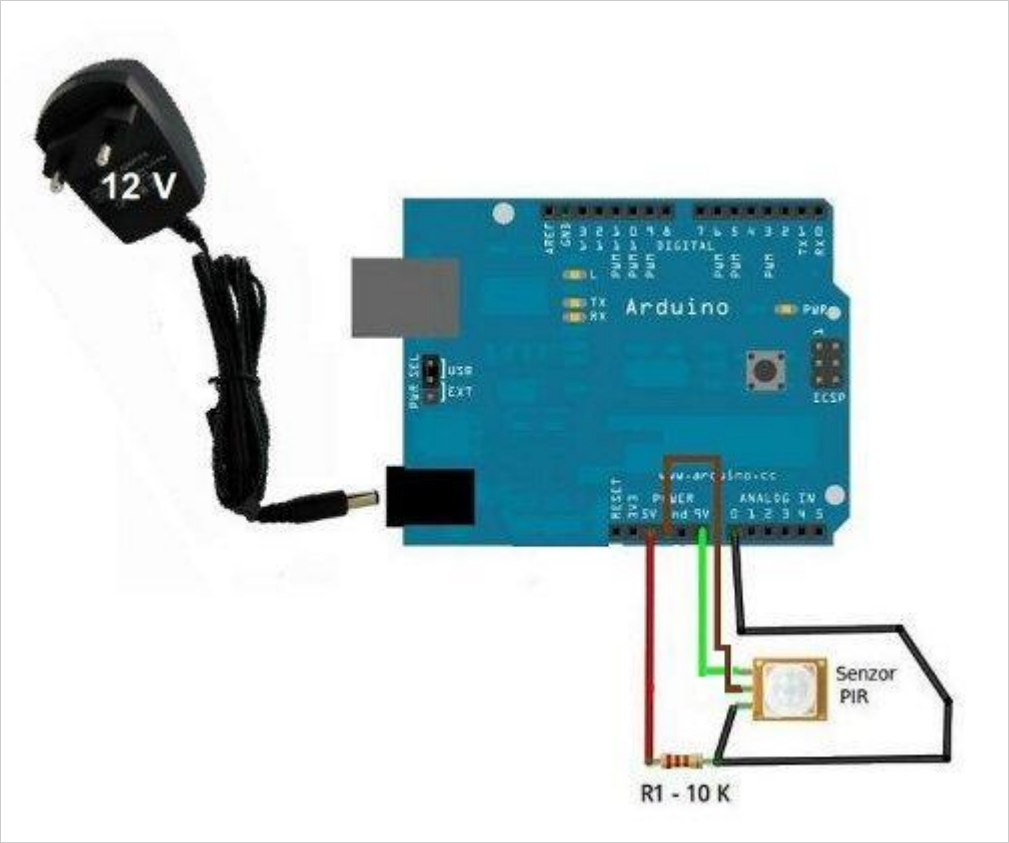
sau restaurantelor ii au pentru a aprinde lumina in mod automat cand intri tu). Functioneaza pe baza detectarii radiatiei infrarosii. Orice corp (si cu atat mai mult oamenii sau animalele) emit radiatie infrarosie in continuu. Elementul de detectie al unui senzor PIR este impartit in doua jumatati egale, iar citirile celor doua jumatati se anuleaza in mod normal (fiind egale). In momentul in care apare insa miscare, una dintre jumatati va detecta mai multa radiatie, iar acest lucru va activa senzorul de prezenta. Raza de detectie a unui senzor PIR poate varia in functie de producator, asa ca va trebui sa citesti datasheet-ul sau sa il testezi. Distanta de detectie cel mai des intalnita este in jurul a cativa metri (3-6 metri).

Senzor PIR

Senzorul PIR este cel mai simplu senzor care detecteaza prezenta umana. Un astfel de senzor are trei fire de conectare. Un fir este VCC (alimentare la 5 V), al doilea este pinul de GND, iar cel de-al treilea este pinul de semnal. Cel mai comun este ca acest fir de semnal sa functioneze in regim "open-collector" (asa functioneaza si senzorul din oferta Robofun – http://www.robofun.ro/senzor_pir). Acest lucru inseamna ca atunci cand senzorul nu detecteaza nimic, pinul de semnal nu este conectat la nimic (este ca si cum ar fi un fir lasat pe birou, neconectat). Atunci cand senzorul detecteaza miscare, pinul de semnal este conectat la GND. Acest comportament este foarte util pentru situatiile in care vrei sa faci o actiune in cazul detectiei de prezenta, fara a folosi un microcontroller. Astfel, este suficient sa conectezi un led inseriat cu un rezistor intre VCC si pinul de semnal, si atunci cand senzorul detecteaza prezenta umana, pinul de semnal este conectat la GND, adica prin led circula curent electric, deci led-ul se aprinde.

Pentru a-l folosi cu Arduino, avem nevoie doar de un rezistor de cativa zeci de Kilo (valoarea exacta nu este prea importanta), si de un alimentator extern de 12 V (sau o baterie de 9V). Cele doua scheme de conectare sunt mai jos.

Atunci cand senzorul nu detecteaza prezenta, firul negru conectat la senzor este ca si cand ar fi lasat pe birou, liber, fara a fi conectat cu nimic. Acest lucru inseamna ca pinul digital 5 al Arduino este tras spre 5V, prin rezistorul de 10K. Atunci cand senzorul detecteaza prezenta, firul negru conectat la senzor este pus la GND, deci pinul digital 5 al Arduino va citi 0 V.



Arduino VIN	Fir Rosu Senzor
Arduino 5V	PIN1 Rezistor 10K
Arduino GND	Fir Maro Senzor
Arduino Pin Digital 5	PIN2 Rezistor 10K
PIN2 Rezistor 10K	Fir Negru Senzor

```
int pirPin = 5;

void setup(){
 Serial.begin(9600);
 pinMode(pirPin, INPUT);
}

void loop(){
 int prezenta = digitalRead(pirPin);
 if(pirVal == LOW){
 Serial.println("Este cineva in camera !");
```

```
      delay(2000);
    }
  }
```

In codul de mai sus, in rutina *setup* pinul digital 5 (cel la care este conectat senzorul) este declarat ca fiind un pin de *INPUT,* iar apoi in rutina *loop*, ori de cate ori acest senzor trece in LOW (0V), inseamna ca senzorul a detectat prezenta.

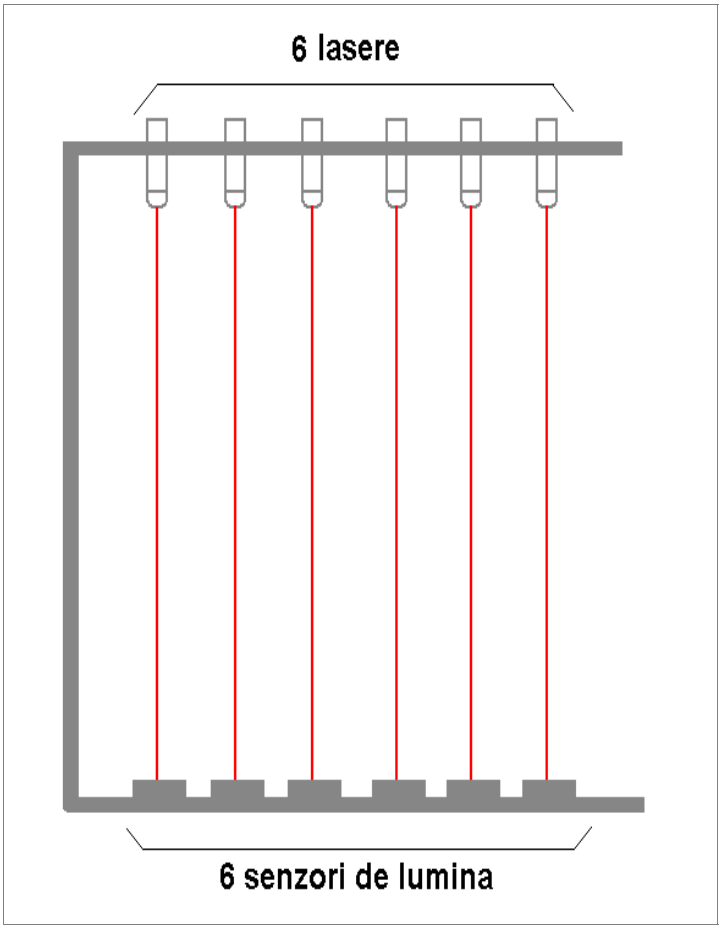
Proiect : Harpa cu laser

Harpa cu laser este un proiect destul de simplu de realizat, bazat pe senzori de lumina pe care cad raze laser. Atunci cand blochezi cu mana una dintre razele laser (similar cu atingerea unei coarde dintr-o harpa reala) este redata o nota muzicala corespunzatoare folosind Music Shield.

Pentru simplitate, in cele ce urmeaza vom alege sa folosim sase senzori de lumina brick si o placa Arduino UNO sau Arduino Leonardo. Cea mai simpla varianta este sa montezi cei sase senzori de lumina pe o bucata de lemn, folosind suruburi cu piulita sau suruburi pentru lemn. Conectarea senzorilor de lumina brick la Arduino o vei face asa cum este descris in sectiunea dedicata acestora (pinul GND la pinul GND al Arduino, pinul VCC la pinul 5V al Arduino, iar pinii de semnal la cate un pin analogic, incepand cu A0 pana la A5).

Music Shield il vei infige pur si simplu in placa Arduino.

Pentru lasere, poti folosi pointer-e laser din comert, cea mai ieftina varianta. Acestea vor fi montate deasupra senzorilor, cate unul deasupra fiecare senzor, astfel incat lumina laser-ului sa fie exact deasupra zonei active a senzorului (fototranzistorul, piesa de culoare galbena). Chiar daca proiectul se cheama "harpa cu laser", laserele sunt mai mult sau mai putin optionale. Daca le montezi, sensibilitatea harpei va fi mai mare, pentru ca lumina generata de laser (si care pica pe senzor) este foarte puternica, si atunci cand o intrerupi cu mana, vei obtine o variatie mare in valoarea citita de senzor. Daca alegi sa nu montezi laserele, in continuare atunci cand misti mana deasupra senzorilor vei obtine o variatie in valoarea citita de senzor (pentru ca vei obtura o parte din lumina naturala care pica pe senzor). Variatia va fi insa mai mica decat atunci cand ai laserele montate si va varia cu distanta la care iti misti mana fata de senzori. Sugestia mea ar fi sa incerci prima data fara lasere si daca nu iti place cum functioneaza, sa le adaugi ulterior.



Arduino va monitoriza cei sase senzori de lumina in mod continuu si atunci cand valoarea citita de unul dintre senzori scade sub un anumit prag prestabilit (din cauza faptul ca am obturat cu mana raza laser), atunci Arduino va comanda Music Shield generarea notei respective.

```
#include <SoftwareSerial.h>

#define PRAG_SENZORI 90
#define DURATA_NOTA 2000

#define INSTRUMENT 7 //(de la 1 la 128)

#define VOLUME 127 // (maxim 127)

SoftwareSerial mySerial(2, 3);

byte note = 0;
byte resetMIDI = 4;
byte ledPin = 13;
static byte senzori[6]= {
  0, 1, 2, 3, 4, 5};

byte triggered[16] = {
  0, 0, 0, 0, 0, 0};

static byte string[6] = {
  60, 62, 64, 65, 67, 68};
```

```
byte hitNote[6] = {
  0, 0, 0, 0, 0, 0};

uint32_t timestamp[6];

void setup() {
  Serial.begin(57600);
  mySerial.begin(31250);
  pinMode(resetMIDI, OUTPUT);
  digitalWrite(resetMIDI, LOW);
  delay(100);
  digitalWrite(resetMIDI, HIGH);
  delay(100);

  for(int i=0; i<6; ++i) {
    timestamp[i] = 0xFFFFFFFF;
  }

  talkMIDI(0xB0, 0x07, VOLUME); //setare volum

  talkMIDI(0xC0, INSTRUMENT, 0); //setare instrument
}



void loop() {
  for(int i=0; i<6; ++i) {
    if((analogRead(senzori[i]) < PRAG_SENZORI) && (triggered[i] == 0)){
      triggered[i] = 1;
      noteOn(0, string[i], 127);
    }
    else if((triggered[i] == 1) && (analogRead(senzori[i]) >=
PRAG_SENZORI)){
      triggered[i] = 0;
      timestamp[i] = millis();
      hitNote[i] = 1;
    }
  }

  for(int i=0; i<6; ++i) {
    if( (millis() > timestamp[i] + DURATA_NOTA) && (hitNote[i] == 1) ) {
      noteOff(0, string[i] ,127);
      hitNote[i] = 0;
    }
  }
}

void noteOn(byte channel, byte note, byte attack_velocity) {
  talkMIDI( (0x90 | channel), note, attack_velocity);
}

void noteOff(byte channel, byte note, byte release_velocity) {
  talkMIDI( (0x80 | channel), note, release_velocity);
}

void talkMIDI(byte cmd, byte data1, byte data2) {
  digitalWrite(ledPin, HIGH);
  mySerial.write(cmd);
```

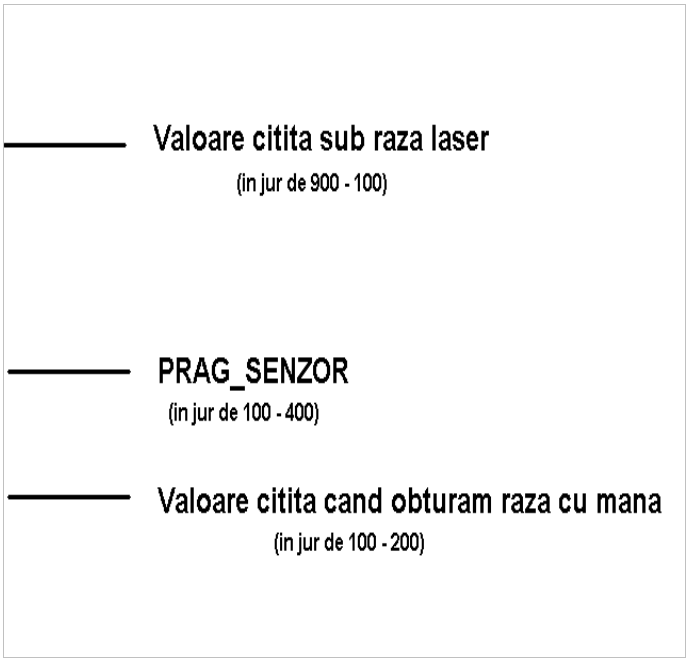
```
    mySerial.write(data1);

    if( (cmd & 0xF0) <= 0xB0) {
      mySerial.write(data2);
    }

    digitalWrite(ledPin, LOW);
  }
```

Constanta PRAG_SENZORI (definita chiar la inceputul programului) stabileste nivelul la care se genereaza o nota. Valoarea acestei constante trebuie sa fie cu putin mai mare decat valoarea citita de senzorul de lumina cand obturam raza laser cu mana. Daca setam pentru aceasta constanta o valoarea prea mare, riscam sa depasim valoarea citita de senzor in mod normal, si atunci notele vor fi generate tot timpul. Daca setam o valoare prea mica, atunci vom ajunge sub valoarea citita atunci cand obturam raza laser cu mana si nu vom reusi sa generam note niciodata. Vezi schema de mai jos pentru mai multa claritate. Cel mai simplu sa obtii o valoare functionala este sa afisezi in interfata de debug valorile citite de senzori atunci cand raza laser lumineaza senzorul si cea atunci cand raza laser este obturata cu mana, si apoi sa setezi o valoare care sa fie cu 50 – 100 mai mare decat valoarea citita atunci cand raza laser este obturata.

Urmatoarea constanta, DURATA_NOTA defineste cat timp se aude o nota dupa ce a fost generata. Cu cat valoarea este mai mica, cu atat notele vor fi mai scurte. Esti liber sa experimentezi cu aceasta valoare cum doresti tu.

Constanta INSTRUMENT defineste ce tip de instrument va reda notele muzicale. Valoarea aleasa in programul de mai sus, 7, corespunza unei harpe. Tu poti alege orice valoare doresti din lista de mai jos. Denumirile instrumentelor sunt in limba engleza (lista a fost extrasa direct din documentul de specificatii pentru Music Instrument Shield), asa ca simte-te liber sa experimentezi si sa alegi ce-ti place mai mult. Cum ar fi oare o harpa care sa sune ca un acordeon ? Acum poti sa afli singur.

VOLUME defineste cat de tare doresti sa se auda notele. Valorile posibile sunt intre 0 si 127.



Un film cu o astfel de harpa laser gasesti aici :

http://www.tehnorama.ro/the-sound-of-shadow-and-light-the-movie/

Aceasta a fost lectia 7. In final, as vrea sa te rog sa ne oferi feedback asupra acestei lectii, pentru a ne permite sa le facem mai bune pe urmatoarele.

Este vorba despre un sondaj cu 4 intrebari (oricare este optionala), pe care il poti accesa dand click aici.

Sau ne poti contacta direct prin email la contact@robofun.ro .

Iti multumim,

Echipa Robofun.RO